

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 60-045855
 (43)Date of publication of application : 12.03.1985

(51)Int.Cl. G06F 12/08
 G06F 3/06
 G11B 20/10

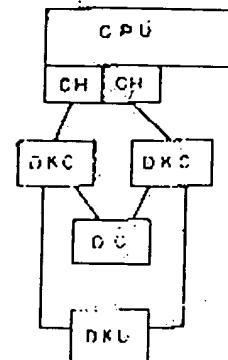
(21)Application number : 58-152859 (71)Applicant : FUJITSU LTD
 (22)Date of filing : 22.08.1983 (72)Inventor : KANEKO SATORU

(54) SEQUENTIAL ACCESS DETECTING METHOD OF MAGNETIC DISK DEVICE

(57)Abstract:

PURPOSE: To ensure the effective use of a cache memory of a large scale sequential access by detecting and deciding a sequential access and then defining a track with a proven an access as a release object at a subsequent time.

CONSTITUTION: A cache memory DC has the capacity to store the data of plural tracks of a disk DKU and also contains a control table for those track data. When a CPU requests the record on a track TR2, a cache memory control mechanism gives -1 to the track number and checks the immediately preceding track TR1 in case the record requested from the CPU does not exist on the memory DC. If the requested record exists on the control table together with the sequential flag and the final record access flag set at 1 respectively, the TR1 passes through by a sequential access. Then it is decided that the sequential access reaches the TR2, and all records of the TR1 on the memory DC and the corresponding rows on the table are released to define the object to be driven out in near future e.g. the next time or the next time but one.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

[decision of rejection]
[Date of extinction of right]

⑯ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A) 昭60-45855

⑬ Int.Cl.

G 06 F 12/08
3/06
G 11 B 20/10

識別記号

庁内整理番号

8219-5B
6974-5B
8322-5D

⑭ 公開 昭和60年(1985)3月12日

審査請求 有 発明の数 1 (全5頁)

⑮ 発明の名称 磁気ディスク装置の順次アクセス検出方法

⑯ 特願 昭58-152859

⑰ 出願 昭58(1983)8月22日

⑱ 発明者 金子悟 川崎市中原区上小田中1015番地 富士通株式会社内

⑲ 出願人 富士通株式会社 川崎市中原区上小田中1015番地

⑳ 代理人 弁理士 青柳 稔

明細書

1. 発明の名称

磁気ディスク装置の順次アクセス検出方法

2. 特許請求の範囲

中央処理装置、多数のトラックを有し各トラックに複数のレコードを有する磁気ディスク装置、及びキャッシュメモリを備え、中央処理装置が或るレコードを要求し、それがキャッシュメモリ上にあれば、キャッシュメモリから転送し、なければ磁気ディスク装置をアクセスして当該レコードがあるトラックを読み出し、該トラックの全レコードをキャッシュメモリへステージングすると共に要求されたレコードを中央処理装置へ転送する磁気ディスクサブシステムにおける順次アクセス検出方法において、

中央処理装置が要求するレコードに関する情報をトラック別に記録するテーブルを設け、該テーブルに中央処理装置から要求があったレコードの番号の最新のものを記録するとともに、このレコードが存在するトラックに対して順次アクセスさ

れたことを示すフラグを立て、

中央処理装置が或るレコードを要求するときそのレコード番号を前記テーブル上のレコード番号と比較して連続性を調べ、連続性が検出されれば該テーブル上の該フラグをセットしたままとし非連続性が検出されれば該フラグは落とし、更にトラック間に跨って順次アクセスされていることを検出すれば先にアクセスされたトラックはキャッシュメモリにおける解放対象に指定することを特徴とした磁気ディスク装置の順次アクセス検出方法。

3. 発明の詳細な説明

発明の技術分野

本発明は、ディスクキャッシュメモリの効率的な使用に用いる磁気ディスク装置の順次アクセス検出方法に関する。

従来技術と問題点

磁気ディスク装置は記憶容量が大きいがアクセスタイムが大であるので半導体メモリで構成されるキャッシュメモリを付設し、ディスクからキャッ

シュメモリへデータをステージングして該キャッシュメモリから中央処理装置(CPU)へデータを送るという方式を採用するメモリシステムがある。ところでキャッシュメモリの記憶容量はディスクと比較すれば遙かに小さいからデータを逐次ステージングして行くと忽ち一杯になってしまう。そこで一杯になつたら、差し当つて必要でないと思われるデータを廃棄し、その跡へ新データを書込むという方式をとるのが普通である。しかしどれを不必要データと判断するかは難しい問題である。一般には古い(書込み時点が早い)データを不必要データとするが、勿論古いデータ即ち不必要データではない。

磁気ディスク装置は一般には多数の同心円状トラックを持ち、各トラックに複数のレコードが書込まれる。そして中央処理装置からのデータ要求はレコード単位であり、そしてあるレコード(R11とする)が要求されると当該レコードが有るトラック(TR1とする)がアクセスされ、該トラックの全レコードが読出されてキャッシュメモ

リへ書込まれ、同時に上記の要求されたレコードR11が中央処理装置へ送られる。中央処理装置が連続的なデータを要求する場合は次の要求レコードはR11に後続するR12であり、レコードR12は上記のようにステージングされてキャッシュメモリ上にあるから、今度はディスクをアクセスすることなく直ちに該キャッシュメモリより読出し中央処理装置へ送ることができる。次のレコードR13が要求される場合も同様であり、こうしてキャッシュメモリを利用すると迅速なデータ読出しが可能になる。

中央処理装置からの要求レコードが次のトラックTR2にあるレコードR21に移ると、該レコードR21はキャッシュメモリ上にはないから再びディスクアクセスが行なわれ、トラックTR2の読出し、キャッシュメモリへのその全レコードの書込み、中央処理装置への要求されたレコードR21の転送が行なわれる。以下同様である。キャッシュメモリの容量は小さいから全トラックのステージングを逐次行なつて行くと忽ちにしてキ

ャッシュメモリは一杯になつてしまい、古いものを廃棄する、の論理で更新するなら古いしかし必要なデータが忽ちにして廃棄されてしまつて、該データが要求される都度ディスクアクセスを行なわざるを得ない結果になる。

発明の目的

本発明は、キャッシュメモリを忽ちにして一杯にしてしまう原因の一つに大規模な連続的レコードの読出し(順次アクセス)にあるとの認識に立ち、そして順次アクセスの場合、あるトラック上のレコードをアクセスすると再びそのレコードをアクセスする確率は極めて低いという経験上の事実により、順次アクセスなら、最後のレコードまで中央処理装置から転送要求のあったトラックのキャッシュメモリ上レコード群は次回等、近未来的廃棄対象としこれによりキャッシュメモリの有効利用を図る、特にかかる制御を可能とするための順次アクセスを検出する方法を提供しようとするものである。

発明の構成

本発明は中央処理装置、多数のトラックを有し各トラックに複数のレコードを有する磁気ディスク装置、及びキャッシュメモリを備え、中央処理装置が或るレコードを要求し、それがキャッシュメモリ上にあれば、キャッシュメモリから転送し、なければ磁気ディスク装置をアクセスして当該レコードがあるトラックを読出し、該トラックの全レコードをキャッシュメモリへステージングすると共に要求されたレコードを中央処理装置へ転送する磁気ディスクサブシステムにおける順次アクセス検出方法において、中央処理装置が要求するレコードに関する情報をトラック別に記録するテーブルを設け、該テーブルに中央処理装置から要求があったレコードの番号の最新のものを記録するとともに、このレコードが存在するトラックに對して順次アクセスされたことを示すフラグを立て、中央処理装置が或るレコードを要求するときそのレコード番号を前記テーブル上のレコード番号と比較して連続性を調べ、連続性が検出されれば該テーブル上の該フラグをセットしたまゝとし

特開昭60-45855(3)

非連続性が検出されれば該フラグは落とし、更にトラック間に跨って順次アクセスされていることを検出すれば先にアクセスされたトラックはキャッシュメモリにおける解放対象に指定することを特徴としたが、次に実施例を参照しながらこれを詳細に説明する。

発明の実施例

第1図はキャッシュメモリを持つディスクサブシステムの構成を示し、CPUは前述の中央処理装置、DKUは磁気ディスク装置、DCはディスクキャッシュメモリである。図面ではDKUは1つのみ示すが、大型システムでは複数個あり、複数のホストCPUがこれらのDKUを使用するのが普通である。キャッシュメモリDCの容量は例えば1MB、トラックで言って200トラック分であるが、複数のホストが共用するので1ホスト当たりでは可成り小容量である。DKCは磁気ディスク制御装置、CHはチャネルで、これらは複数用いられ、どの系統からもディスクDKU及びキャッシュメモリDCをアクセスできるようにされる。

ディスク制御装置DKCはディスクDKUを制御し、データをDKUからCHへまたCHからDKUへ転送する。またそれと同時にキャッシュメモリDCに対してデータ転送を行なう。ディスクDKUから読出したデータをキャッシュメモリDCに複写することをステージングと呼び、これは前述のようにCPUから要求されたデータがキャッシュメモリDCにない場合にトラック単位で行なわれる。CPUから要求されたデータがDC上に存在する場合は、DKUをアクセスすることなく該DCよりCPUへ当該データを転送する。

キャッシュメモリDCはDKUの複数のトラックのデータを格納できる容量を持ち、そのトラックデータ（こゝでは単にトラックともいう）を管理するテーブルを持っている。DCが一杯になったとき、更にトラックを格納するには該DC内の適宜のトラックを解放し、新らしいトラックをステージングする余地を作るが、これに通常用いられる論理はLRU(Least Recently Used)アルゴリズムであり、最も古くアクセスされたトラッ

クが解放される。しかしこの方法ではDC容量を越えるトラック本数が一度にアクセスされるとDC上の他の全トラックが追い出され、残るは今回アクセス分のみとなってしまう。実際の計算機システムではこれは主に大規模順次アクセスによつてもたらされる。従って本発明では順次アクセスを検出し、当該アクセスが順次アクセスと分るとアクセス済みトラックは次回の解放対象とする。このようにすれば大規模順次アクセスでもキャッシュメモリの2トラック分の記憶領域を交互使用するだけとなり、古いが必要なデータをCDから追放することが避けられる。

本発明の順次アクセス検出要領を第2図、第3図で説明するに、第2図は連続した2つのトラックTR1、TR2の形式を示し、△印はトラックの基点、R11、R12、……は前述のレコードを示す。第3図はキャッシュメモリを管理するテーブルを示し、図示のように本発明ではトラックアドレス、シーケンシャルフラグ、最終レコードアクセスフラグ、最終レコード番号の各項目から

なる行の複数個で構成する。各行L1～Lnの配列順はLRUアルゴリズムによる。「トラックアドレス」は、磁気ディスクのトラックのアドレスであり、CPUがアクセス時に発行したものである。「シーケンシャルフラグ」は、CPUのディスクアクセスがあり当該ディスクのトラックがDCにステージングされた時点で一応（この時点ではシーケンシャルか否かは分らないが）セットされる。この時同時にCPUが要求したレコードの番号が「最終アクセスレコード番号」の項目に格納される。勿論この時点では「最終アクセス」か否か分らないが、アクセス毎に更新するから結果的に最終アクセスレコード番号を表示することになる。

CPUアクセスに対する処理は先ずDC上で実行され、該レコード番号が調べられる。上記のDC適所に格納したレコード番号がR11、今度のCPUアクセス対象のレコード番号がR12であると、アクセスはシーケンシャルであるからシーケンシャルフラグはリセットしない。しかし今回

のCPUアクセスレコード番号がR13などであるとアクセスはシーケンシャルでないからシーケンシャルフラグをリセットする。いずれの場合も当該レコード番号R12またはR13などを「最終アクセスレコード番号」へ格納する。若し今回CPUアクセスレコード番号がR21など他のトラックにあるものであると、ディスクアクセス等が行なわれ、第3図のテーブルでは新たな行が起され(トラックTR2はまだアクセスされていないとして)、当該行に所要事項が書込まれる。また「最終レコードアクセスフラグ」はトラックの最後のレコードがアクセスされたときセットされる。

今トラックTR1のレコードR11がアクセスされ、それはキャッシュメモリDC上に存在しないとすると、ディスクがアクセスされ、トラックTR1の全レコードが読出されてキャッシュメモリDCへステージングされかつそのうちのレコードR11がチャネルCHを通してCPUへ送られる。また管理テーブルにはトラックTR1の行が

起され、トラックアドレス=1、シーケンシャルフラグ=1、最終レコードアクセスフラグ=0、最終アクセスレコード番号=1となる。CPUが続いてレコードR12を要求すると、これは先程記憶した最終アクセスレコード番号(=1)に+1したものと等しいから順次アクセスであると認められ、シーケンシャルフラグはそのまま1、最終アクセスレコード番号は2に更新される。またレコードR12はDCからCPUへ転送される。次のCPU要求レコードはR13であるとすると、これは2+1に等しいからやはり順次アクセスと認められて上記と同様な処理が行なわれ、そしてR13はトラック最終レコードとするとDCのテーブルでは最終レコードアクセスフラグが1にセットされる。

次にトラックTR2上のレコードが要求され、これはDC上に存在しない場合を考える。キャッシュメモリ制御機構はトラック番号を-1してその直前のトラック本例ではTR1の状態を調べ、それがテーブル上にありかつシーケンシャルフラ

グ=1、最終レコードアクセスフラグ=1であれば、トラックTR1は順次アクセスで通過し更に該順次アクセスはトラックTR2に入ってきたと判断してDC上のTR1の全レコードおよびテーブル上の当該行を解放する(次回または次々回等、近未来に追い出す対象とする)。

順次アクセスはトラックTR1, TR2, TR3, ……と多数トラックに亘って行なわれることがしばしばあるから、上記のトラックTR2のレコードが順次アクセスされたとき当該トラックTR2の全レコードだけでなく次のトラックTR3の全レコードもDCにプリステージングすると、予想通り該TR3のレコードも順次アクセスされるとき、アクセス時間減少に効果的である。このような場合は上記のように解放対象としたTR1のDC上格納領域にTR3を格納するとよい。

発明の効果

以上説明したように本発明によれば大容量の順次アクセスでもキャッシュメモリ上の記憶領域を殆んど消費しないのでキャッシュメモリ上の他のデ

ータを追い出さず、このためヒット率(DC上にデータがある割合)を高めることができ、高速アクセスを保証することができる。また順次アクセスを検出することによりプリステージングまたはブリッヂが可能となり、順次アクセスに対するディスクシステムの性能を向上させることができる。

4. 図面の簡単な説明

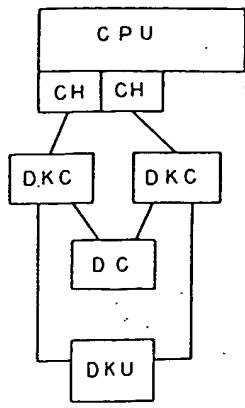
第1図はキャッシュメモリ付き磁気ディスクメモリシステムの構成を示すブロック図、第2図はディスク上のレコードの説明図、第3図は本発明で用いるテーブルの説明図である。

図面でCPUは中央処理装置、DKUは磁気ディスク装置、DCはキャッシュメモリ、TR1, TR2, ……はトラック、R11, R12, ……はレコードである。

出願人 富士通株式会社
代理人弁理士 背柳稔

第1図

特開昭60-45855(5)



第3図

	トラック アドレス	シーケンシ アルフラグ	最終レコード アクセスフラグ	最終アクセス レコード番号
L1				
Ln				

第2図

TR1 ▽ [R11] [R12] [R13] ▽
TR2 ▽ [R21] [R22] [R23] ▽